#247 Cifre Ord 1

Se dau mai multe numere naturale, fiecare cu cel mult *9* cifre. Să se afişeze, în ordine descrescătoare, toate cifrele care apar în numerele date.

Fişierul de ieşire *cifreord1.out* va conţine cifrele determinate, ordonate descrescător, câte *20* pe o linie, valorile de pe fiecare linie fiind separate prin spaţii. Ultima linie a fişierului poate conţine mai puţin de *20* de cifre.

#include <fstream> using namespace std;

ifstream fin("cifreord1.in"); ofstream fout("cifreord1.out");

int fr[10], x, i;

short b;

int main()

{

while(fin>>x)

do

{

fr[x%10]++;

x/=10;

}while(x>0);

b=20;

for(i=9;i>=0;i--)

while(fr[i]>0)

{

fout<<i<<' ';

fr[i]--;

b--;

if(b==0)

{

fout<<endl;

b=20;

}

}

return 0;

}

*cifreord1.in*

301941 81912 83392

776996 431446

*cifreord1.out*

9 9 9 9 9 8 8 7 7 6 6 6 4 4 4 4 3 3 3 3

2 2 1 1 1 1 1 0

#SumChef – 1306

# Cerința Fiind dat un şir cu *n* elemente numere naturale, să se calculeze suma cuburilor cifrelor tuturor numerelor din şir. Date de intrare Programul citește de la tastatură numărul *n* şi cele *n* elemente ale şirului. Date de ieșire Programul va afișa pe ecran numărul *S*, reprezentând suma cuburilor cifrelor tuturor numerelor din şir .

|  |  |
| --- | --- |
| #include <iostream>  using namespace std;  int n,x,i,c,fr[10];  long long s=0;  int main()  {  cin>>n;  for(i=1;i<=n;i++)  {  cin>>x;  while(x>0)  {  c=x%10;  x=x/10;  fr[c]++;  }  }  for(i=1;i<=9;i++)  s=s+fr[i]\*i\*i\*i;  cout<<s;  return 0;  } | #include <iostream>  using namespace std;  int n,x,i,c,fr[10];  long long s=0;  int main()  {  cin>>n;  for(i=1;i<=n;i++)  {  cin>>x;  while(x>0)  {  c=x%10;  x=x/10;  fr[c]++;  }  }  for(i=1;i<=9;i++)  s=s+fr[i]\*i\*i\*i;  cout<<s;  return 0;  } |
| 90p | 100p |

#1005 Numere 8

Se dă o listă cu numere naturale. Să se determine numerele naturale nenule cu cel mult patru cifre care nu apar în lista dată.

#include <fstream>

using namespace std;

ifstream fin("numere8.in");

ofstream fout("numere8.out");

int n,m;

int a[10000];

int main()

{

while(fin>>n)

{

if(n<10000)

a[n]=1;

}

for(int i=9999;i>=1;i--)

if(!a[i])

fout<<i<<" ";

return 0;

#525 Numere 1

Se dau *n* numere naturale. Determinaţi cele mai mari două numere cu trei cifre care nu apar printre numerele date.

#include <iostream> using namespace std; int fr[1000],n;

int main() { int i,x,j,y; cin>>n;

for (i=1;i<=n;i++) {cin>>x; if (x<=999) fr[x]++; }

x=999;

while (x>99 && fr[x]>0)

x--;

y=x-1;

while (y>99 && fr[y]>0)

y--;

if (y>99)

cout<<y<<' '<<x;

else

cout<<"NU EXISTA";

}

#187 Cif Frecv

Să se scrie un program care citeşte cel mult *1000000* de numere naturale din intervalul închis *[0,9]* şi determină cel mai mare număr prim citit şi numărul său de apariţii.

#include <fstream>

using namespace std;

ifstream fin("ciffrecv.in");

ofstream fout("ciffrecv.out");

int mx, a, x;

int main()

{

mx=0;

while(fin>>x)

{

if(x==2||x==3||x==5||x==7)

{

if(x>mx)

{

mx=x;

a=1;}

else

if(x==mx)

a++;}}

fout<<mx<<' '<<a;

return 0;

}

#include <fstream> using namespace std;

int v[11],i,x,a,b;

ifstream fin("ciffrecv.in"); ofstream fout("ciffrecv.out");

int main()

{

while(fin>>x)

v[x]++;

for(x=7;x>=3;x=x-2)

if(v[x]) { a=x; break; }

if(a==0 &&v[2]) a=2;

fout<< a<<' '<<v[a];

return 0;

}

#239 Nr lipsa

Se dau mai multe numere naturale cu cel mult *9* cifre fiecare. Să se determine cele mai mari două numere de *3* cifre care nu se găsesc printre numerele date.

#include <fstream> using namespace std;

int main() {

fstream cin("nrlipsa.in", ios::in), cout("nrlipsa.out", ios::out);

bool x[1000] = {0};

long a, m = 0, q = 0;

while(cin >> a)

if(a < 1000)

x[a] = true;

for(int i = 999;i >= 100;i--) {

if(!x[i])

m = q,

q = i;

if(m && q) break;

}

if(m && q)

cout << m << " " << q;

else

cout << "NU";

return 0;

}

#264 Max cif

Se dau mai multe numere numere naturale formate din exact o cifră. Determinaţi cifrele cu număr maxim de apariţii.

#include <fstream> using namespace std;

int main() {

fstream fin("maxcif.in", ios::in), fout("maxcif.out", ios::out);

long a, x[10] = {0}, max = 0;

while(fin >> a)

x[a]++;

for(int i = 0;i < 10;i++)

if(x[i] > max)

max = x[i];

for(int j = 0;j < 10;j++)

if(x[j] == max)

fout << j << " ";

return 0;

}